

VŠB – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky

BAKALÁŘSKÁ PRÁCE

2011

Jakub Misiarz

VŠB – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

Stereo korespondence
Stereomatch

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

v Ostravě dne 22.7. 2011

Podpis:

Poděkování

Touto cestou bych rád poděkoval svému vedoucímu, Ing. Michalu Krumníkovi, za cenné rady a odbornou pomoc při realizaci této práce. Dále bych rád poděkoval svým blízkým za podporu během celého studia.

Abstrakt

Tato bakalářská práce se zabývá problémem stereo korespondence. Jedná se o hledání všech shodných párů pixelů v obrazech, zobrazujících jednu scénu z více úhlů pohledů. Cílem této práce je seznámení s problémy spojené se zpracováním obrazů a tvorby disparitní mapy. V současnosti existuje mnoho metod, které řeší problém stereo korespondence. V této práci jsou popsány principy vybraných metod, jejich implementace a testování. Jedná se o metody SAD, SSD, ZSSD, ZSAD a SHD.

Klíčová slova

Stereo korespondence, Disparitní mapa, OpenCV, SAD, SSD, ZSSD, ZSAD, SHD, Census transformace

Abstract

This bachelor thesis deals with the stereo correspondence problem. This problem can be described as finding all pairs of identical pixels in images, which shows a scene from multiple angles. The aim of this work is to introduce problems related to image processing and creation of disparity map. Currently, there are many methods to solve the stereo correspondence problem. In this work are described principles of choosen methods, their implementation and testing. These are methods SAD, SSD, ZSSD, ZSAD and SHD.

Keywords

Stereomatch, Stereo correspondence, Disparity map, OpenCV, SSD, SAD, ZSSD, ZSAD, SHD, Census transformation

Seznam použitých zkratek a symbolů

SSD	Sum of Squared Differences
SAD	Sum of Absolute Differences
ZSSD	Zero Sum of Squared Differences
ZSAD	Zero Sum of Absolute Differences
SHD	Sum of Hamming Distances
CTW	Census Transform Window

Obsah

1 Úvod.....	1
2 Zpracování obrazů	2
2.1 Snímání.....	2
2.2 Digitalizace	2
2.2.1 Vzorkování.....	3
2.2.2 Kvantizace.....	4
2.3 Pořízení obrazů	5
3 Disparitní mapa.....	6
4 Algoritmy stereo korespondence	7
4.1 Census transformace.....	8
4.2 Lokální metody	9
4.2.1 SSD(Sum of Squared Differences).....	10
4.2.2 SAD(Sum of Absolute Differences).....	10
4.2.3 ZSSD(Zero mean Sum of Squared Differences).....	11
4.2.4 ZSAD(Zero mean Sum of Absolute Differences).....	11
4.2.5 SHD(Sum of Hamming Distance).....	12
4.3 Globální Metody	12
5 Implementace	144
5.1 OpenCV	144
5.2 Kontrola parametrů	144
5.3 Tělo funkce	144
6 Testování	166
6.1 SSD.....	177
6.2 SAD	188
6.3 ZSSD	200
6.4 ZSAD.....	211
6.5 SHD.....	222
6.6 Srovnání.....	255
7 Závěr.....	266
8 Literatura	277

1 Úvod

Počítačové vidění je obor výpočetní techniky, který se zabývá získáváním informací ze zachyceného obrazu. Jedná se např. o rozpoznávání registračních značek automobilů a obličejů, zpracování videa nebo analýzu materiálů.

Stereo korespondence tradičně byla a je jedním z nejdiskutovanějších témat počítačového vidění. Jedná se o hledání všech shodných párů pixelů v obrazech, zachycujících jednu scénu z více úhlů pohledu.

Zrak je pro člověka nejdůležitější smysl. 80% ze všech získaných informací zajišťuje právě zrak. Je to smysl, který člověku umožňuje vnímat tvary, světlo a různé barvy. Vnímání prostoru je umožněno polohou očí, kdy do každého oka dopadá mírně odlišný obraz, z nichž se v mozku skládá prostorový obraz prostředí. Tuhle schopnost se člověk naučí už jako malé dítě.

Počítače a výpočetní technika dokážou člověka nahradit v mnoha ohledech, včetně zraku. Prostorové vidění je zejména důležité pro robotiku, kde je potřeba, aby se roboti bezpečně pohybovali v neznámém prostředí. Technicky můžeme lidské oko nahradit kamerou, popřípadě fotoaparátem pro získání dvourozměrných obrazů a lidský mozek procesorem. Aby však robot dokázal prostorově vnímat, je nutné nejdříve navrhnout metodu, která zpracuje vstupní obrazy. Naším problémem tedy pro modelování počítačového vidění je rekonstrukce trojrozměrného obrazu ze dvou obrazů dvourozměrných. Při výběru metody je důležité, aby zpracovala vstupní obrazy nejen kvalitně, ale i rychle. Tato technika se používá nejen v robotice, ale také v dopravě nebo pro zobrazování povrchu planet z družic a satelitů.

Během pořízení a zpracování vstupních obrazů nastává řada problému. V následující kapitole se s těmito problémy seznámíme a popíšeme možné způsoby jejich řešení. Zejména se zaměříme na pořízení obrazů a jejich správnou úpravu.

Poté se zaměříme na metody stereo korespondence. Protože existuje mnoho různých algoritmů, které řeší problém stereo korespondence, je potřeba si tyto metody rozdělit a u vybraných metod popsat jejich principy. Následně se budeme zabývat implementací těchto metod. V kapitole 6 pak tyto metody otestujeme na vstupních obrazech a zhodnotíme výsledky jednotlivých metod.

2 Zpracování obrazů

Člověk je pomocí svých smyslů schopen vnímat objekty, jevy a procesy reálného světa. Prostřednictvím barev, tvarů, zvuků a vůně přijímá data z jeho okolí. Na druhé straně počítače nejsou schopny vnímat svět kolem nás jako my. Jsou schopny přijímat a pracovat s daty pouze v binárním kódu. Přesto počítače dokážou zobrazit obrazy a přehrát video nebo hudbu. Aby byly schopny toto provést, musíme data z reálného světa, tak jak je vnímáme my, převést do podoby srozumitelné počítači, tedy do binárního kódu.

Zpracování a rozpoznávání obrazu reálného světa rozdělujeme do několika základních kroků [2]:

- Snímání a digitalizace obrazu
- předzpracování
- segmentace obrazu
- popis objektů
- klasifikace

V této kapitole se budeme věnovat pouze pro nás nejdůležitějším krokem a to je snímání a digitalizace obrazu. Na závěr této kapitoly si popíšeme jakým způsobem správně pořídit obrazy pro další zpracování a proč je to důležité.

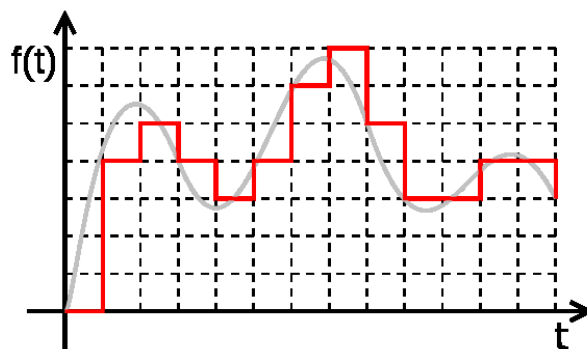
2.1 Snímání

Snímáním obrazu rozumíme převod optické veličiny na elektrický signál. Tento proces můžeme přirovnat k radiometrickému měření. Na výsledný obraz má vliv mnoho faktorů jako osvětlení snímaného objektu nebo jeho samotné vlastnosti, např. odrazivost povrchu.

2.2 Digitalizace

Jelikož počítače pracují s daty pouze v binárním kódu, musíme převést získané analogové data na digitální. Tento proces se nazývá digitalizace a skládá se ze dvou kroků: *vzorkování(sampling)* a *kvantizace(quantization)*.

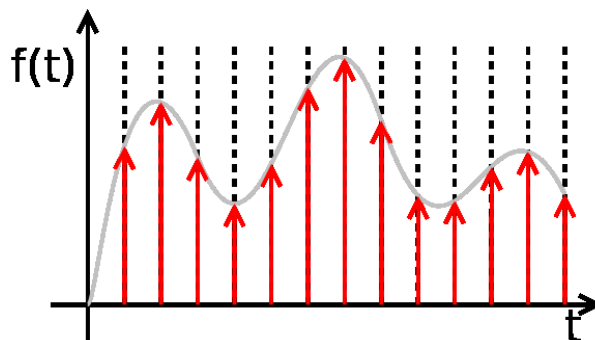
Digitální obraz je získán vzorkováním obrazu do matice $M \times N$ bodů a kvantování do K úrovní [3]. Na obrázku 2.1 je zobrazen výsledný signál po vzorkování i kvantizaci.



Obr 2.1 Digitalizace analogového signálu

2.2.1 Vzorkování

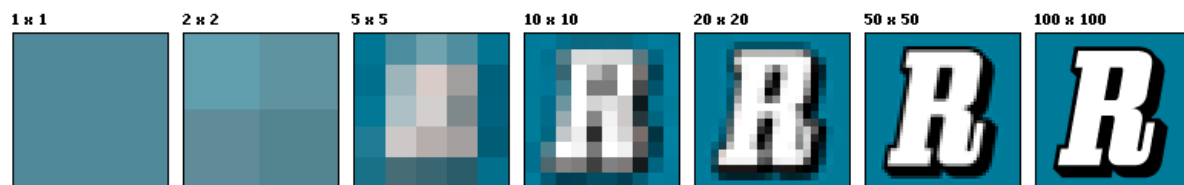
Vzorkováním se zjišťují hodnoty(vzorky) v pravidelných intervalech. Původní analogový signál je přetvořen na posloupnost okamžitých hodnot, viz obr 2.2.



Obr 2.2 Vzorkování signálu

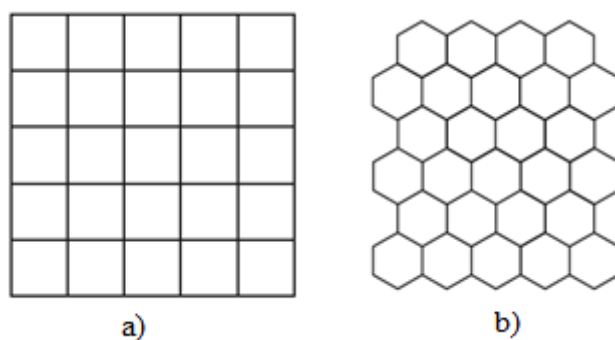
Jednomu vzorkovacímu bodu odpovídá v obraze jeden pixel(po digitalizaci). Pixel je plocha konečné nenulové velikosti. Tuto plochu prezentuje jediná konstantní hodnota, obvykle získaná ze středu pixelu.

Velmi důležité je zvolit správné rozlišení. Při nízkém dochází ke ztrátám v obraze, naopak při vysokém rozlišení získáme detailní obraz, ale za cenu zvýšené náročnosti při dalším zpracování obrazu(obr 2.3).



Obr 2.3 Rozdíly v rozlišení

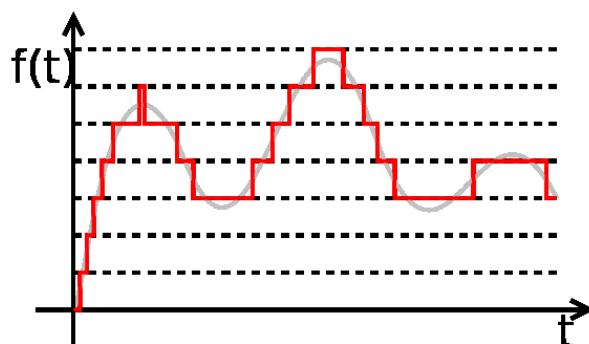
Dalším důležitým krokem je výběr vzorkovací mřížky. Nejčastěji používané jsou mřížky čtvercové a hexagonální (obr 2.4). Výhodou čtvercové mřížky je snadná realizace, avšak oproti hexagonální mřížce má nevýhody jako měření vzdáleností a spojitostí pixelů. Hexagonální mřížka zase není vhodná pro operace, jako je např. Fourierova transformace. Podrobnější informace lze nalézt v [1].



Obr 2.4 Čtvercová(a) a hexagonální(b) mřížka

2.2.2 Kvantizace

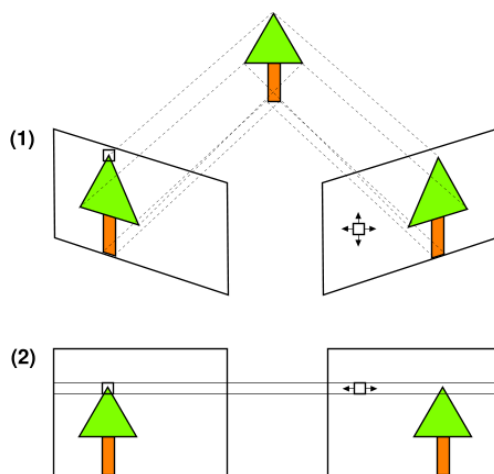
Kvantizací aproximujeme obor hodnot spojitě funkce. Je to proces, při kterém se obor hodnot funkce rozdělí na intervaly, kterým je přidělena jedna zástupná hodnota (obr 2.5)



Obr 2.5 Kvantizovaný signál

2.3 Pořízení obrazů

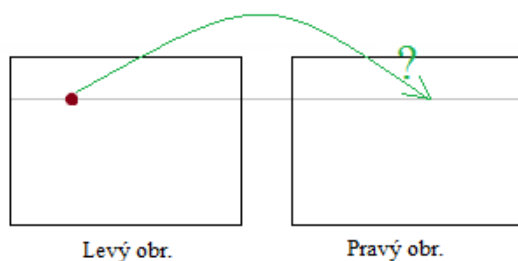
Pro zachycení obrazů zobrazující jednu scénu z více úhlů se používá jedna pohybující se kamera nebo statická soustava dvou a více kamer. Důležitá součást pro další zpracování obrazů je především kalibrace kamer. Snímky pořízené z nekalibrovaných kamer jsou výpočetně náročnější na zpracování, protože řádky na kterém se nachází hledaný pixel se nemusí shodovat. Jelikož většina metod stereo korespondence, včetně těch, které implementujeme v této práci potřebuje pro svůj správný chod, aby si hledané pixely řádkově odpovídaly, musíme buď vstupní obrazy rektifikovat(obr 2.6) nebo správně kalibrovat kamery.



Obr 2.6 Obrázky před(1) a po (2) rektifikaci

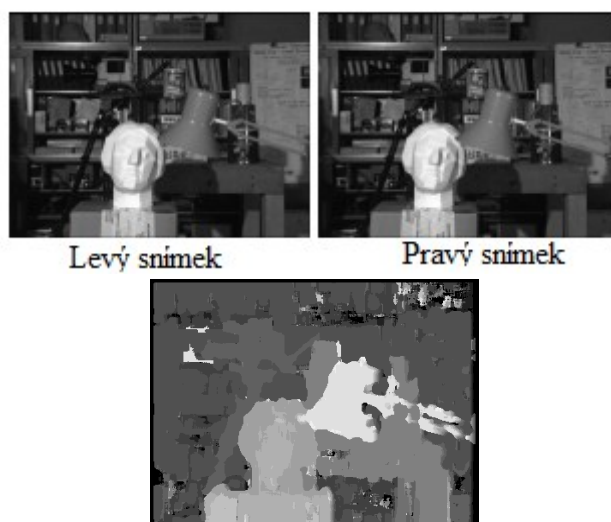
3 Disparitní mapa

Z předchozích kapitol již víme, že hledání shodných parů pixelů v obrazech probíhá v jednom odpovídajícím řádku. Cílem je zjistit, kde se hledaný pixel posunul, viz obr. 3.1. Pro tuto vzdálenost byl zaveden pojem disparita. Všechny pixely, jejichž pozice na řádku si odpovídají, mají disparitu nulovou. Cílem metod stereo korespondence je tvorba disparitní mapy, která obsahuje odhad disparity pro každý pixel.



Obr 3.1 Porovnání pixelu

Disparitní mapa je matice o rozměrech $M \times N$, kde M je výška a N šířka vstupních obrazů. Hodnoty jednotlivých polí matice odpovídají vzdálenosti pixelu o poloze x, y v referenčním obraze od jeho korespondujícího pixelu v obraze druhém. Z těchto hodnot jsme následně schopni určovat reálné vzdálenosti objektů. Na obrázku 3.2 jsou zobrazeny vstupní obrazy (Levý snímek, Pravý snímek) a výsledná disparitní mapa.



Obr 3.2 Ukázka disparitní mapy

4 Algoritmy stereo korespondence

Ačkoli do dnešní doby bylo vyvinuto mnoho algoritmů stereokorespondence, většina z nich se chová velmi podobně. Ikdyž mohou mít rozdílné vlastnosti, během výpočtu používají funkce s podobným významem. Jejich principem je určit podobnost intenzit(jasu) odpovídajících si pixelů pro různé disparity. Pokud si intenzity pixelů odpovídají, obsahuje disparitní mapa nulovou hodnotu a naopak, čím víc se hodnoty jasů pixelů liší, tím větší hodnotu bude obsahovat. Účelem daného algoritmu je vybrat co nejideálnější disparitu, tak aby odpovídala rozmístěním objektů v obraze. Jedná se tedy o výběr té disparity, při které se intenzity pixelů nejvíce shodovaly.

Způsob určování podobnosti intenzit pixelů a výběr nejoptimálnější disparity, je pro každý z algoritmů stereokorespondence specifický.

Společnými rysy jsou následující 4 fáze výpočtu:

1. Výpočet podobnosti pixelů pro všechny disparity
2. Sloučení ohodnocení pro danou disparitu
3. Tvorba disparitní mapy
4. Úprava a doladění výsledné disparitní mapy.

V první fázi měříme podobnost mezi dvěma pixely, respektive okolí dvou pixelů. Pro výpočet podobnosti se používá např. squared intensity differences (SD – kvadrát rozdílů jasu), absolute intensity differences (AD – absolutní hodnota rozdílů jasu) nebo Hamming Distance (HD – bitová nonekvivalence).

Ve druhé fázi se slučují vypočtené podobnosti. U lokálních metod se jedná např. o součet nebo průměr okolí pixelu(okénka). Tato fáze slouží k vyloučení jedno pixelových chyb a k tvorbě souvislejší plochy.

Ve třetí fázi sestavujeme výslednou disparitní mapu. Existují dva rozdílné přístupy pro tento krok – lokální a globální. Zatímco lokální metody pracují jen s okénkem daného pixelu, globální metody hledají nejlepší disparitu po celém řádku na kterém se pixel nachází.

Výpočet výsledné disparity v lokálním přístupu probíhá jednoduše – vítězná disparata je ta, při které si byla daná okolí pixelů(okénka) nejvíce podobná. Tomuto principu se říká „winner takes all“(WTA).

Algoritmus globálních metod se dá popsat jako nalezení minima funkce ohodnocení. Jedná se o nalezení disparitní funkce d , která minimalizuje globální energii (viz rovnice 4.1)[6].

$$E(d) = E_{data}(d) + \lambda E_{smooth}(d)$$

Rovnice 4.1 Funkce minimalizující globální energii [6]

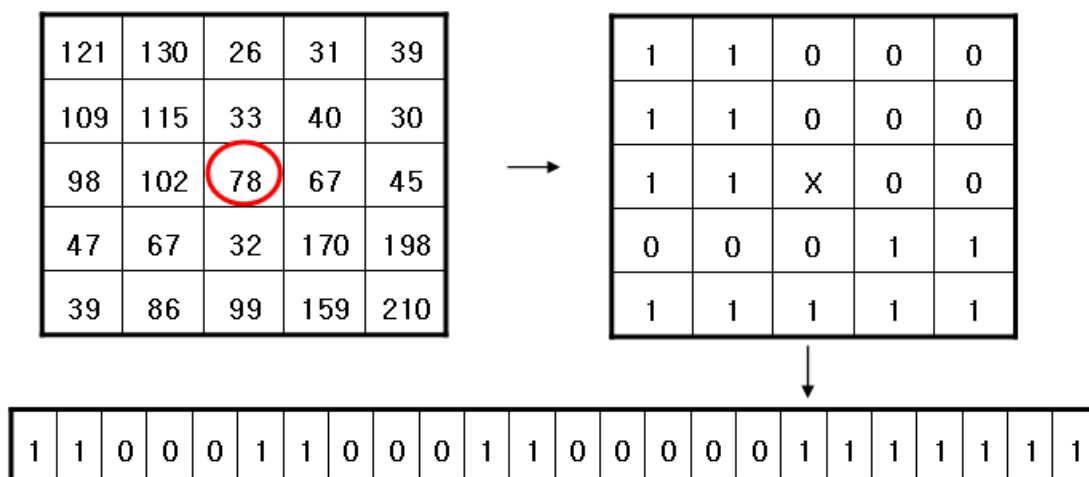
V poslední fázi upravujeme disparitní mapu do finální podoby. Tato fáze je zaměřena především na vylepšení kvality výsledné disparitní mapy. Nejčastěji se jedná o doplnění disparit do míst, ve kterém nebyla určena nebo například použití filtru k rozostření objektů v obraze. Další způsoby vylepšení kvality jsou popsány v [7].

V této práci se zabýváme a implementujeme některé z lokálních metod, proto se teď na ně zaměříme podrobněji. Vysvětlíme si jejich principy a popíšeme jejich výhody a nevýhody. Na závěr kapitoly si pak ve zkratce objasníme principy globálních metod.

Nejdřív se ale podíváme na census transformaci obrazu, kterou využívá metoda SHD.

4.1 Census transformace

Census transformace je forma lokální, neparametrické transformace, která se užívá ve zpracování obrazů k mapování hodnoty intenzity pixelu uvnitř čtvercového okolí na bitový řetězec, čímž zachycuje strukturu obrazu [8]. Hodnota intenzity centrálního pixelu, ve zkoumaném okénku, je nahrazena bitovým řetězcem, který vznikne porovnáváním ostatních intenzit pixelu v okénku s centrálním pixelem. Porovnávání probíhá řádkově, zleva do prava (obr 4.1). Pokud je intenzita zkoumaného pixelu v okénku větší než intenzita centrálního pixelu, do bitového řetězce se přidá 1, pokud není, tak 0.



Obr 4.1 Census transformace okénka [9]

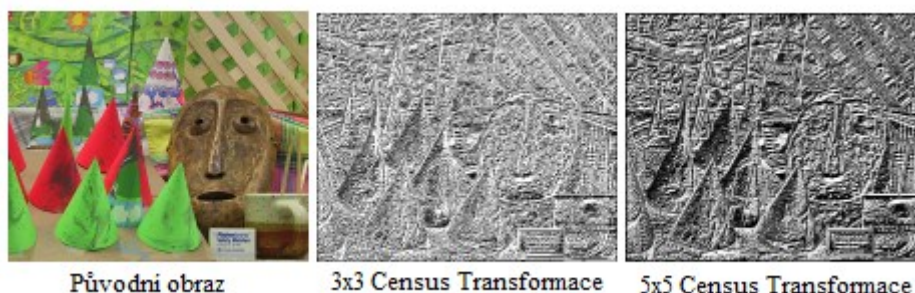
Výhody Census transformace:

- Redukuje výkyvy způsobené špatným nastavením kamer

- Kóduje lokální prostorovou strukturu

Nevýhody:

- Ztráta původní informace o pixelu
- Když se zvětší okénko CTW(velikost okénka census transformace), zvětší se také požadavky na velikosti proměnných. Například pro okénko 3×3 potřebujeme proměnnou pro uložení census hodnoty velikosti 2^3 neboli 8 bitů, pro CTW o velikosti 5×5 už je to $2^5 = 32$ bitů.



Obr 4.2 Příklady census transformace

4.2 Lokální metody

Výpočet výsledné disparitní mapy je, jak už bylo zmíněno výše, jednoduchý. Pro každý pixel se vybere disparita s nejmenším ohodnocením, tedy ta disparita, pro kterou si byly daná okénka pixelů nejvíce podobná.

Hlavním problémem těchto metod je správné zvolení velikosti okénka(okolí kolem pixelu). Rozměr okénka by se měla volit tak, aby pokryl dostatek pixelů pro správné vyhledávání a zároveň tak, aby nedošlo k efektům promítacího narušení. Pokud zvolíme malé okénko, ve výsledné disparitní mapě budou špatné odhady disparit, protože okénko pokrylo málo pixelů a algoritmus neměl dostatek variací jasu pro správné párování. Pokud zvolíme okénko příliš velké, můžeme pokrýt i pixely určující hloubku(pixelů viditelné pouze v jednom z obrazů) a výsledek bude nekvalitní také. Navíc se zvýší náročnost na výpočet výsledné disparity.

Dalším důležitým problémem je volba disparit, tedy vzdálenosti na řádku, pro které se má hledat okénko s nejmenším ohodnocením. Pokud disparitu zvolíme příliš velkou, stoupne časová náročnost výpočtu. Pokud příliš malou, výsledná disparitní mapa bude nepřesná.

Jelikož v následující části budeme pro každou uvedenou metodu popisovat rovnici ohodnocení, vysvětlíme si nejdříve jednotlivé symboly v rovnicích:

Symbol	Popis
$U = U(x,y)$	Okolí kolem bodu daného souřadnicemi (x,y)
$I_k(x,y)$	Hodnota jasu pixelu o souřadnicích (x,y), kde k je číslo daného obrazu.
$\overline{I_k(x,y)} = \frac{1}{N} \sum_{(i,j) \in U} I_k(x+i, y+j)$	Průměrná hodnota jasu v daném okolí okolo pixelu o souřadnicích (x,y), kde k je číslo obrazu
dx, dy	Disparity ve směru x,y

Tabulka 4.1 Vysvětlení symbolů a výrazů

4.2.1 SSD(Sum of Squared Differences)

Jedná se o jeden z velmi jednoduchých algoritmů, přesto poskytuje relativně slušné výsledky.

Princip této metody spočívá v tom, že se v referenčním obraze porovnávají hodnoty jasu ve zvoleném okénku okolo pixelu, s hodnotami jasu v korespondujícím obraze pro danou disparitu, podle rovnice 4.2. Čím menší hodnotu dostaneme, tím menší jsou rozdíly jasu mezi zvolenými okénky.

$$D_{SSD} = \sum_{(i,j) \in U} \left(I_1(x+i, y+j) - I_2(x+d_x+i, y+d_y+j) \right)^2$$

Rovnice 4.2 Vzorec pro ohodnocení shody metodou SSD [7]

4.2.2 SAD(Sum of Absolute Differences)

Metoda SAD je velmi podobná metodě SSD, akorát při ohodnocení shody nepoužívá součet mocnin rozdílů jasů pixelů, ale součet absolutních rozdílů jasů, viz rovnice 4.3. Stejně jako v případě SSD platí, že čím menší hodnotu ohodnocení získáme, tím jsou rozdíly jasu

pixelů menší a shoda pravděpodobnější. SAD je méně náročná na výpočet, proto dostává přednost před SSD v aplikacích, kde záleží na rychlosti výpočtu.

$$D_{\text{SAD}} = \sum_{(i,j) \in U} |I_1(x + i, y + j) - I_2(x + d_x + i, y + d_y + j)|$$

Rovnice 4.3 Vzorec pro ohodnocení shody metodou SAD [7]

4.2.3 ZSSD(Zero mean Sum of Squared Differences)

Metoda ZSSD vychází z metody SSD, ale při ohodnocování pixelu odečítá průměrný jas daného okénka od každého pixelu v okénku, viz rovnice 4.4.

ZSSD je časově náročnější na výpočet, ale výsledná disparitní mapa je kvalitnější než u SSD.

$$D_{\text{ZSSD}} = \sum_{(i,j) \in U} \left[[I_1(x + i, y + j) - \overline{I_1(x, y)}] - [I_2(x + d_x + i, y + d_y + j) - \overline{I_2(x + d_x, y + d_y)}] \right]^2$$

Rovnice 4.4 Vzorec pro ohodnocení shody metodou ZSSD [7]

4.2.4 ZSAD(Zero mean Sum of Absolute Differences)

Metoda ZSAD vychází z metody SAD, ale stejně jako ZSSD při ohodnocování pixelů odečítá průměrný jas daného okénka od každého pixelu v okénku, viz rovnice 4.5. Je výpočetně rychlejší než ZSSD a výsledné disparitní mapy by měly být přesnější než u metody SSD.

$$D_{\text{ZSAD}} = \sum_{(i,j) \in U} \left| [I_1(x + i, y + j) - \overline{I_1(x, y)}] - [I_2(x + d_x + i, y + d_y + j) - \overline{I_2(x + d_x, y + d_y)}] \right|$$

Rovnice 4.5 Vzorec pro ohodnocení shody metodou ZSAD [7]

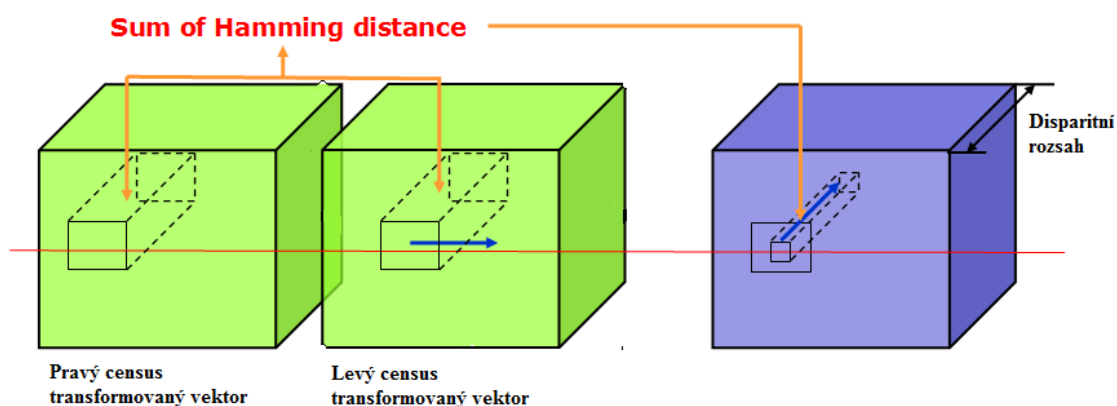
4.2.5 SHD(Sum of Haming Distance)

Tato metoda se normálně používá pro census-transformovány obrazy(viz kapitola 4.2), ale může pracovat i s obrazy bez této úpravy. Princip této metody je založený na počítání bitové nonekvivalence(XOR) hodnot levého a pravého obrazu v daném okénku.

Zatímco u výše zmíněných metod se v referenčním obraze porovnávaly hodnoty jasy ve zvoleném okénku okolo pixelu, s hodnotami jasu v korespondujícím obraze, u metody SHD se neporovnávají jasy ale bitové řetězce, viz rovnice 4.6.

$$D_{SHD} = \sum_{(i,j) \in U} I_1(i,j) \text{ bitwise XOR } I_2(x+i, y+j)$$

Rovnice 4.6 Vzorec pro ohodnocení shody metodou SHD [7]



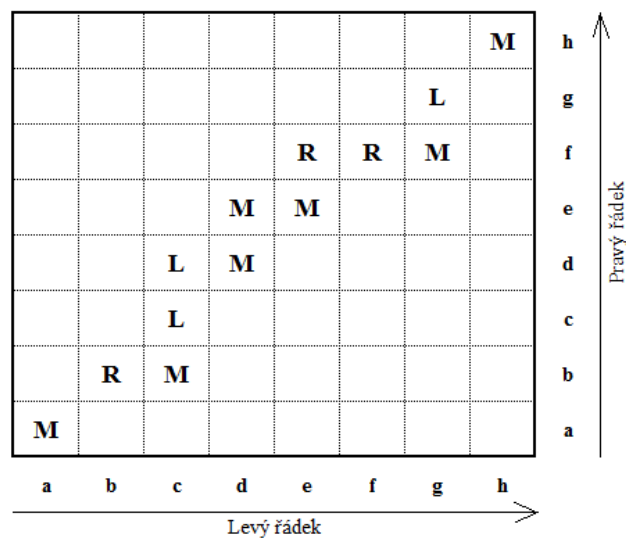
Obr 4.3 Znázornění metody SHD [9]

4.3 Globální Metody

Globální metody sice nemají tak lehkou implementaci jako lokální, ale zato výsledné disparitní mapy obsahují méně chyb a jsou kvalitnější. Globální metody pracují lépe s překrytými objekty než lokální metody, které takové objekty neberou v úvahu. Dále globální

metody dosahují dobrých výsledků i v obrazech, kde je málo textur. Více o výhodách a nevýhodách globálních metod lze nalézt v [7].

Na počátku této kapitoly o algoritmech jsme si popsali jeden z možných algoritmů globálních metod. Další možný přístup ke globální optimalizaci může být např. dynamické programování. Principem je vytvoření matice z korespondujících řádků a následné hledání minimální cesty (viz obr 4.4). Písmena *a-h* označují intenzity jednotlivých řádků. Písmena *L, M, R* označují hledanou cestu. Pixely, které si odpovídají jsou označeny písmenem *M*. Písmenem *L* jsou označeny ty pixely, které jsou viditelné pouze v levém obraze a písmenem *R* zase pixely viditelné pouze v obraze pravém.



Obr 4.4 Dynamické programování [5]

Další možné přístupy ke globální optimalizaci jsou (více v [7] kapitola 6.5).:

- Propagace zpráv (*Belief propagation*)
- Grafové řezy (*Graph cut*)
- Maximální tok sítí (*Max-flow*)
- Nelinerární rozptyl (*Nonlinear diffusion*)
- Tensor hlasování (*Tensor voting*)
- Hierarchické metody (*Hierarchical methods*)

5 Implementace

V této kapitole popíšeme postup implementace našich vybraných metod stereo korespondence. Jelikož všechny implementované metody mají stejnou stavbu a liší se jen ve způsobu ohodnocování pixelů, vysvětlíme implementaci na jednom modelu. Nejdříve si ale popíšeme knihovnu OpenCV, kterou v programu používáme pro práci s obrazy.

5.1 OpenCV

OpenCV(Open-source Computer Vision) je knihovna funkcí, která slouží ke zpracování obrazů v reálném čase. Obsahuje funkce počítačového vidění i strojového rozpoznávání předmětů. Jelikož je volně šiřitelná, umožňuje studentům i profesionálům jednoduše realizovat obtížné projekty.

5.2 Kontrola parametrů

Výsledný program je konsolová aplikace, která pro spuštění vyžaduje několik parametrů ve správném tvaru. Konkrétně je potřeba zadat název levého, pravého obrazu, název výsledné disparitní mapy, velikost okénka, minimální a maximální disparitu a název zvolené metody. Postupně tedy kontrolujeme, zda je zadán dostatečný počet parametrů, jestli vstupní obrazy existují a pokud ano, tak jestli si odpovídají rozměrově. Dále testujeme zda zadané okénko je liché číslo, jestli je minimální disparita menší než maximální a nakonec, zda název metody odpovídá některé z implementovaných.

5.3 Tělo funkce

Než přikročíme k ohodnocování pixelů, převedeme vstupní obrazy na černobílé, protože tak lépe získáme hodnotu intenzity daného pixelu. Dále si vytvoříme prázdný obraz o rozměrech vstupních obrazů, který nám bude sloužit jako disparitní mapa. Poté si určíme, který ze vstupních obrazů bude referenční, tzn. obraz pro jehož pixely budeme hledat shodné, respektive nejvíc podobné pixely v korespondujícím obraze. Předtím než začneme procházet obrazy po jednotlivých řádcích a sloupcích, musíme si uvědomit jeden problém, který může nastat. Jedná se o problém, kdy zvolené okénko zasahuje mimo obraz(viz obrázek 5.1). Tento problém řešíme tak, že při procházení obrazem přeskočíme hraniční řádky a sloupce, ve kterých okénko zasahuje mimo obraz. Např. pokud zvolené okénko má velikost 3, a tím pádem okolí kolem pixelu je 1, první řádek a sloupec který procházíme bude druhý. To platí také o posledním procházeném řádku a sloupci, s tím rozdílem, že posledně procházený řádek a sloupec nebude

druhý, ale předposlední(viz obr 5.1, v červeném rámečku jsou sloupce a řádky, které se budou procházet). Tento problém také může nastat v případě, když zvolené okénko pro danou disparitu přesáhne šířku obrazu. Tento problém řešíme stejně jako v předchozím případě, tzn. sloupce pro které okénko zasahuje mimo obraz přeskočíme.

	80	86	80	90	90
	90	100	90	100	100
	8	90	100	86	100
	90	86	100	100	90
	100	100	86	100	100
	86	90	90	100	100

Obr 5.1 Řešení problému s okénkem zasahujícím mimo rozsah obrazu

Cílem metody je najít k pixelu z referenčního obrazu shodný nebo nejvíce podobný pixel z korespondujícího obrazu. Pro každý pixel na řádku z referenčního obrazu zkoumáme jeho okolí a porovnáváme s okolím korespondujícího obrazu, posunutého o danou disparitu(obr 5.2)

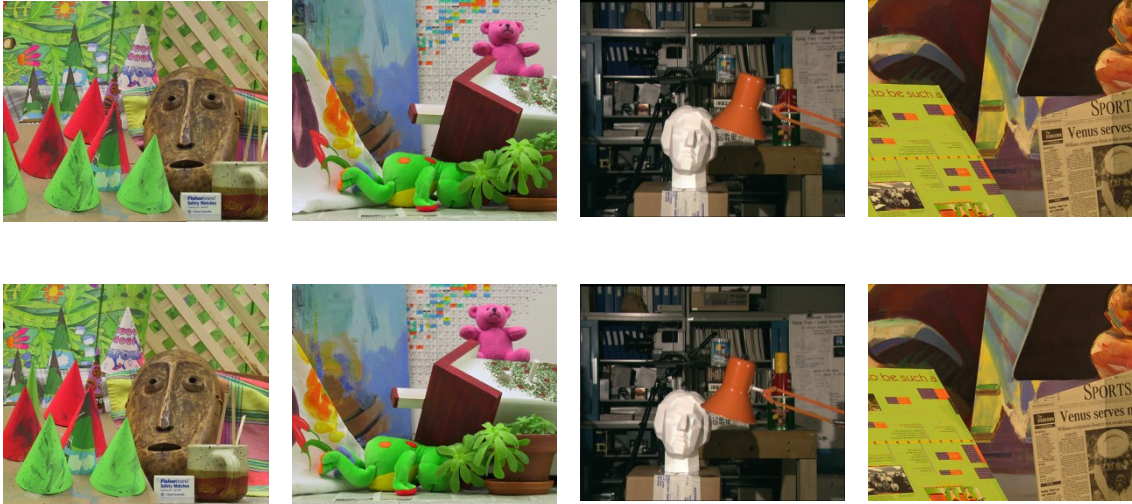
Jednotlivé pixely v okénku na stejných pozicích se ohodnotí podle konkrétní metody a podobnost okének se uloží do proměnné. Pokud je podobnost pro danou disparitu větší(rozdíly okének je menší) než pro předchozí, je tato disparita nastavená jako disparita se zatím nejlepší shodou pixelů. Poté co ohodnotíme všechny okénka dostupná pro zadanou maximální disparitu, hodnotu vítězné disparity uložíme do obrazu disparitní mapy na pozici zkoumaného pixelu z referenčního obrazu.

1	1	1	1	1	1	2	4	2	3	1	1	2	4	2	3	1	1	2	4	2	3	1	1
2	2	2	2	2	2	1	3	2	4	2	2	1	3	2	4	2	2	1	3	2	4	2	2
3	3	3	3	3	3	1	1	2	3	3	2	1	1	2	3	3	2	1	1	2	3	3	2
2	4	5	4	8	7	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
5	4	4	5	4	4	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
						d = 0						d = 1						d = 2					
Referenční obraz						Korespondující obraz																	

Obr 5.2 Příklad hledání nejvíce podobného okénka

6 Testování

Metody, které implementujeme v programu otestujeme na sadě stereo-párů(obr 6.1), které jsou volně dostupné na stránkách <http://vision.middlebury.edu/stereo/data/>. Tato stránka se zabývá metodami stereo korespondence, zkoumá jejich výkon a kvalitu.

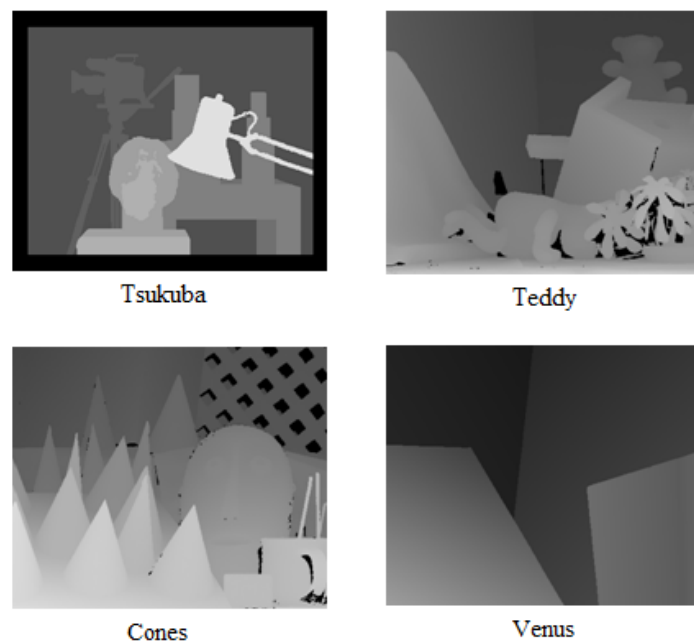


Obr 6.1 Stereo páry, na kterých byly otestovány implementované metody, zleva: Cones, Teddy, Tsukuba, Venus

Naše výsledné disparitní mapy jsme pak porovnávali se vzorovými disparitními mapami(obr 6.2) podle vzorce 6.1[6]. Proměnné d_c a d_t určují disparitu na pozici x,y ve výsledné a vzorové disparitní mapě, δ značí toleranci pro chybu a N celkový počet pixelů. Jelikož disparitní mapy na těchto stránkách jsou zobrazené na větší subpixelovou přesnost, před porovnáváním jsme si naše disparitní mapy museli upravit tak, že jsme vynásobili disparity doporučeným měřítkem uvedeným na těchto stránkách. Pro testování jsme použili scény s názvem *Cones*, *Teddy*, *Tsukuba* a *Venus*. Testovali jsme rozdíly výsledné disparitní mapy při různém nastavení velikosti maximální disparity a zvoleného okénka. Pro každou metodu jsme provedli testování zvlášť a nejlepší výsledky každé metody jsme pak navzájem porovnali.

$$B = \left(\frac{1}{N} \sum_{(x,y)} |d_c(x,y) - d_t(x,y)| > \delta \right)$$

Rovnice 6.1 Vzorec pro výpočet vadných pixelů[6]



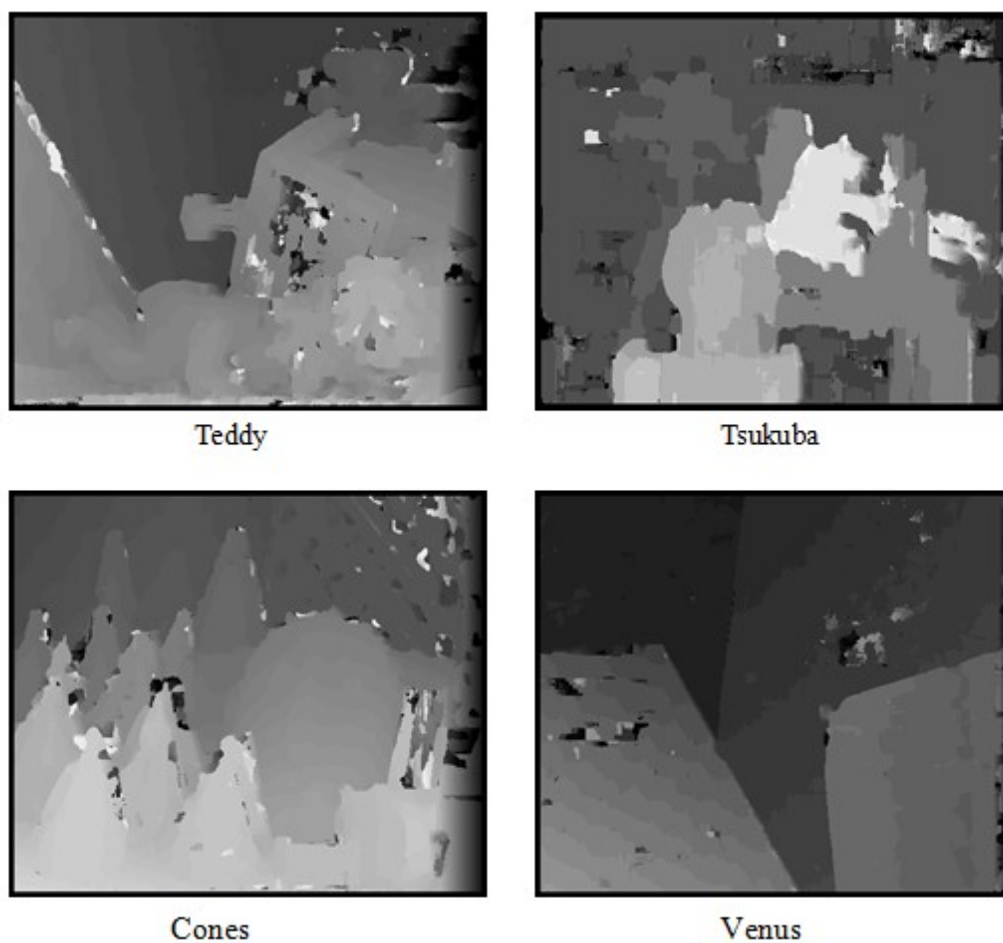
Obr 6.2 Vzorové disparitní mapy

6.1 SSD

Název Scény	Maximální disparita	Velikost okénka	Vadné pixely [%]	Čas [sec]
Tsukuba	15	7	75,7	7,4
Tsukuba	15	9	74,0	8,6
Tsukuba	15	11	72,7	20,2
Cones	59	7	38,8	46,6
Cones	59	9	37,6	78,4
Cones	59	11	37,2	112,0
Teddy	59	7	52,8	46,7
Teddy	59	9	52,4	100,4
Teddy	59	11	52,4	113,6
Venus	19	7	54,9	20,8
Venus	19	9	54,3	26,2
Venus	19	11	54,0	51,9

Tabulka 6.1 Výsledky testování metody SSD

Jak můžeme z tabulky 6.1 vyčíst, metoda SSD je poměrně časově náročná. Nejlepší výsledky byly dosaženy u scény *Cones*, nejhorší u scény *Tsukuba*. Naměřené časy jednotlivých scén nemá smysl navzájem porovnávat, protože u každé scény se použila jiná maximální disparita. Výskyt vadných pixelů klesal u každé scény s narůstající velikostí okénka. Výjimka nastala u scény *Teddy*, kde pro zvolená okénka 9 a 11, zůstalo procento vadných pixelů stejné. Výsledné disparitní mapy, s nejmenším počtem vadných pixelů, jsou na obrázku 6.3.



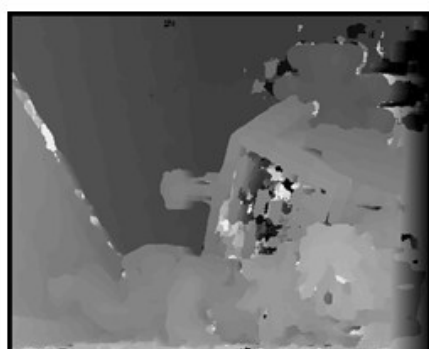
Obr. 6.3 Disparitní mapy vytvořené metodou SSD

6.2 SAD

Disparitní mapy byly vytvořené metodou SAD výrazně rychleji než metodou SSD, čímž jsme potvrdili teorii v kapitole 4. Nejlepší výsledky byly dosaženy u scény *Cones*, nejhorší u scény *Tsukuba*. Stejně jako u SSD platí: čím větší zvolené okénko, tím méně vadných pixelů.

Název Scény	Maximální disparita	Velikost okénka	Vadné pixely [%]	Čas [sec]
Tsukuba	15	7	77,2	2,2
Tsukuba	15	9	76,0	3,7
Tsukuba	15	11	74,9	5,3
Cones	59	7	40,9	11,6
Cones	59	9	38,9	18,5
Cones	59	11	37,7	28,3
Teddy	59	7	52,9	11,8
Teddy	59	9	52,6	18,5
Teddy	59	11	52,6	27,7
Venus	19	7	55,1	4,3
Venus	19	9	54,5	6,9
Venus	19	11	54,2	10,4

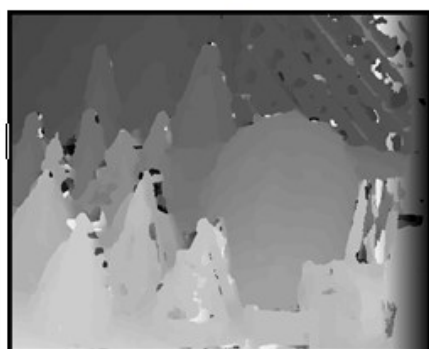
Tabulka 6.2 Výsledky testování metody SAD



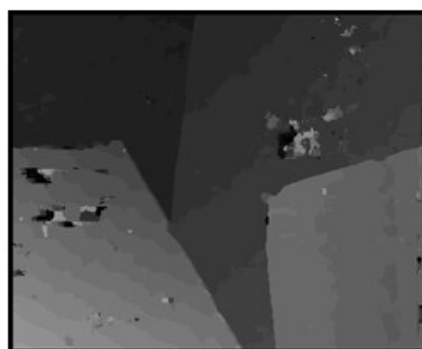
Teddy



Tsukuba



Cones



Venus

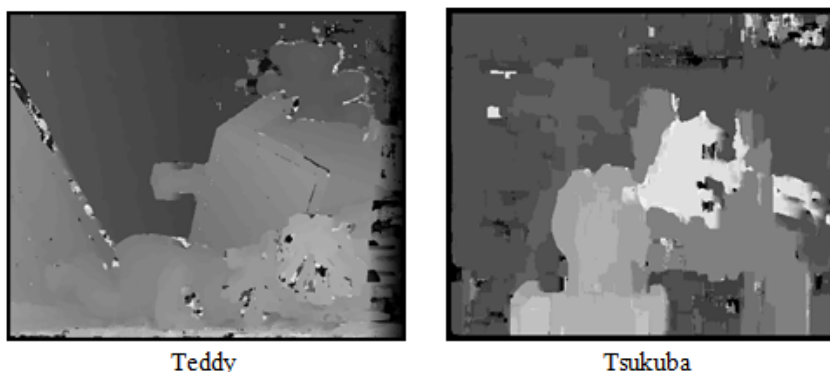
Obr. 6.4 Disparitní mapy vytvořené metodou SAD

6.3 ZSSD

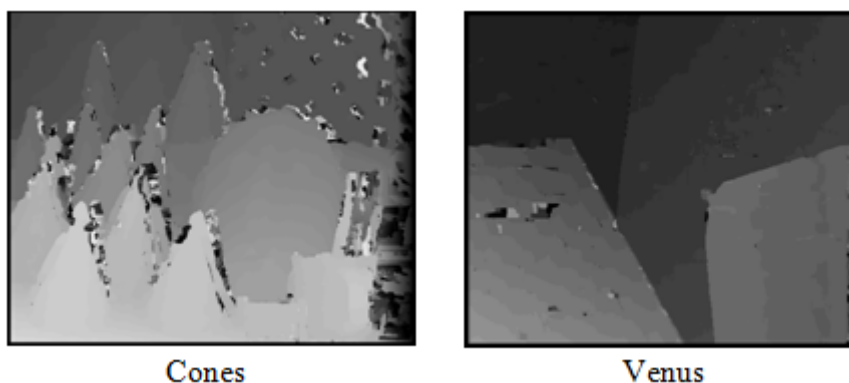
V kapitole 4.1.3 je uvedeno, že metoda ZSSD vychází z metody SSD a zpřesňuje její výsledky. Testováním jsme tuto teorii potvrdili, ale výsledky se zlepšily jen nepatrně. Nejlepší výsledky byly dosaženy u scény *Cones*, nejhorší u scény *Tsukuba*. Během testování scény *Cones* a *Teddy* se při přechodu velikosti okénka z hodnoty 9 na 11, výsledky nezlepšily. Ve zbývajících scénách platí pravidlo, čím větší velikost okénka, tím menší počet vadných pixelů. Časově je tato metoda nejpomalejší ze všech testovaných metod. Výsledné nejlepší disparitní mapy (s nejmenším počtem vadných pixelů), jsou zobrazeny na obrazech 6.5 a 6.6.

Název Scény	Maximální disparita	Velikost okénka	Vadné pixely [%]	Čas [sec]
Tsukuba	15	7	75,6	11,0
Tsukuba	15	9	73,8	17,5
Tsukuba	15	11	72,3	25,8
Cones	59	7	37,7	63,9
Cones	59	9	37,6	97,5
Cones	59	11	37,7	138,4
Teddy	59	7	52,2	59,8
Teddy	59	9	51,9	95,7
Teddy	59	11	52,0	140
Venus	19	7	54,4	19,6
Venus	19	9	53,8	32,7
Venus	19	11	53,6	47,6

Tabulka 6.3 Výsledky testování metody ZSSD



Obr. 6.5 Disparitní mapy vytvořené metodou ZSSD



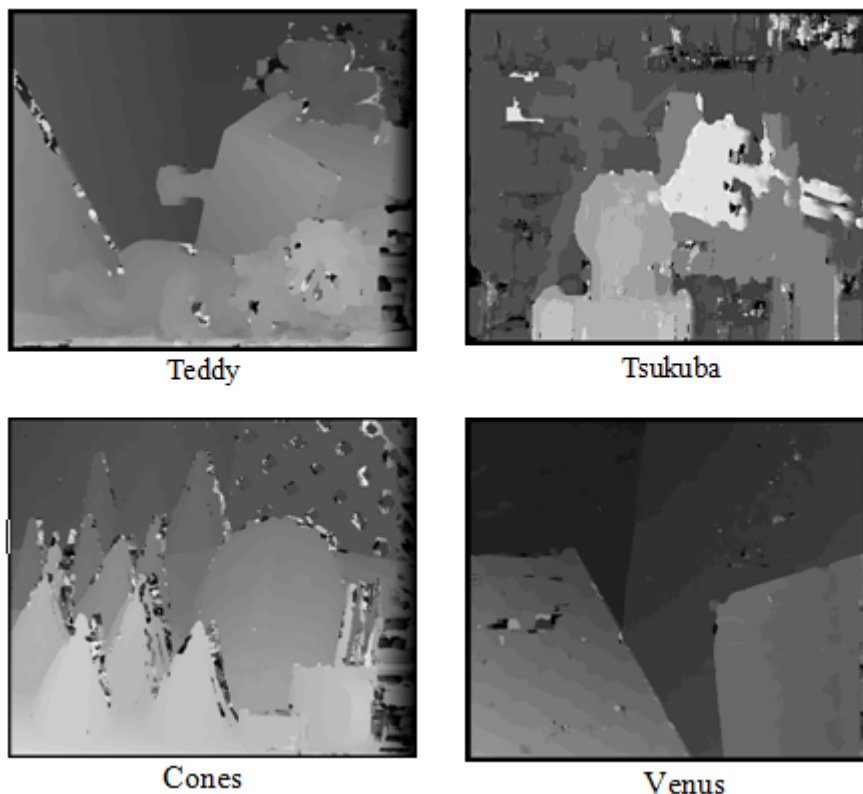
Obr. 6.6 Disparitní mapy vytvořené metodou ZSSD

6.4 ZSAD

Metoda ZSAD vychází z metody SAD a podle teorie v kapitole 4.1.4, by měly být výsledné disparitní mapy přesnější než u metody SAD. Porovnáním tabulek 6.2 a 6.4 zjistíme, že zmíněná teorie platí pro všechny scény. Pravidlo, čím větší velikost okénka, tím přesnější výsledky, platí u této metody pouze ve scéně *Venus* a *Tsukuba*. Ve scéně *Cones* je dokonce pro okénko s největší hodnotou 11, nejhorší výsledek. Z testování jsme dále zjistili, že metoda ZSAD dosahuje kvalitnějších výsledků než ZSSD u scén *Cones* a *Teddy* a je výrazně rychlejší.

Název Scény	Maximální disparita	Velikost okénka	Vadné pixely [%]	Čas [sec]
Tsukuba	15	7	77,0	4,0
Tsukuba	15	9	75,8	6,4
Tsukuba	15	11	74,7	9,4
Cones	59	7	37,4	24,5
Cones	59	9	37,3	36,2
Cones	59	11	37,5	51,4
Teddy	59	7	52,1	21,9
Teddy	59	9	51,9	35,6
Teddy	59	11	51,9	51,6
Venus	19	7	54,5	7,6
Venus	19	9	54,0	12,5
Venus	19	11	53,8	18,1

Tabulka 6.4 Výsledky testování metody ZSAD



Obr. 6.7 Disparitní mapy vytvořené metodou ZSAD

6.5 SHD

Metoda SHD se normálně používá pro census-transformované obrazy (viz kapitola 4.2), ale může pracovat s obrazy i bez této transformace. V tabulce 6.5 jsou zaznamenány výsledky testování obrazů bez census úpravy a v tabulce 6.6 s census úpravou.

Velikost okénka census transformace (CTW) jsme zvolili pouze 3, protože pro jiné velikosti CTW byly obrazy po transformaci nevhodné pro hledání shodných pixelů.

Proto abychom mohli správně porovnat časy testování s census transformací a bez ní, do výsledného času pro testování s census transformací jsme nezapočítávali čas transformace samotné.

Z tabulek 6.5 a 6.6 je patrné, že pro oba testování platí: čím větší velikost okénka, tím kvalitnější výsledky. Výsledné časy u obou testování jsou téměř identické a disparitní mapy s menším počtem vadných pixelů pozorujeme u testování s census transformací. Ve všech scénách kromě scény *Tsukuba* dosahovala metoda SHD s census upravenými obrazy kvalitnější výsledky. Na obrázku 6.8 jsou výsledné nejlepší disparitní mapy obou testování porovnány.

Název Scény	Maximální disparita	Velikost okénka	Vadné pixely [%]	Čas [sec]
Tsukuba	15	7	79,0	1,6
Tsukuba	15	9	77,7	2,5
Tsukuba	15	11	76,5	3,6
Cones	59	7	45,7	8,8
Cones	59	9	42,8	13,3
Cones	59	11	41,0	19,6
Teddy	59	7	55,8	8,5
Teddy	59	9	54,7	13,3
Teddy	59	11	54,1	19,7
Venus	19	7	57,6	2,9
Venus	19	9	56,0	4,6
Venus	19	11	54,8	6,8

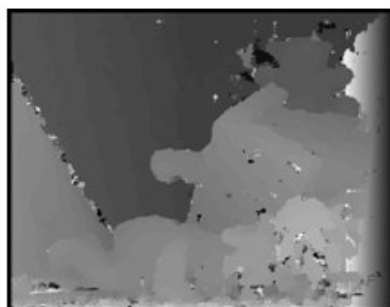
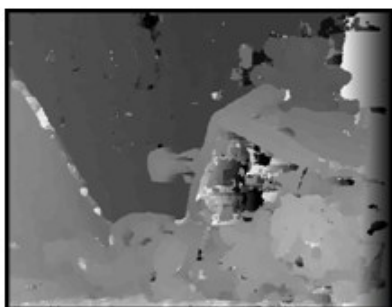
Tabulka 6.5 Výsledky testování metody SHD bez Census Transformace

Název Scény	Maximální disparita	Velikost okénka	Vadné pixely [%]	Čas [sec]
Tsukuba	15	7	84,7	1,5
Tsukuba	15	9	84,2	2,5
Tsukuba	15	11	84,1	3,6
Cones	59	7	37,7	8,2
Cones	59	9	36,6	13,4
Cones	59	11	36,1	19,6
Teddy	59	7	54,3	8,4
Teddy	59	9	52,7	13,3
Teddy	59	11	52,2	19,6
Venus	19	7	57,6	2,8
Venus	19	9	55,8	4,6
Venus	19	11	54,8	6,8

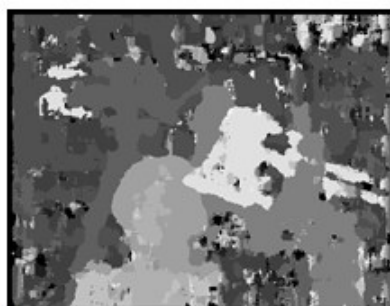
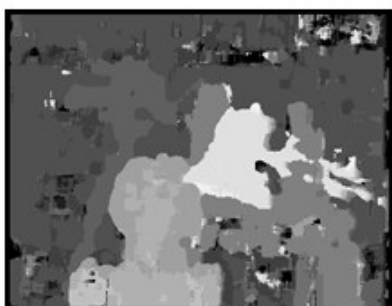
Tabulka 6.6 Výsledky testování metody SHD s Census Transformací

SHD bez Census Transform

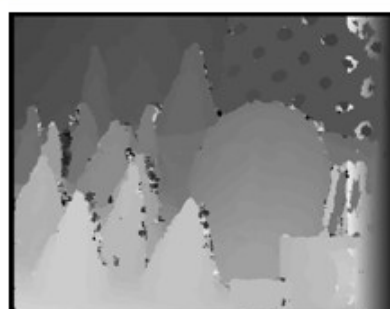
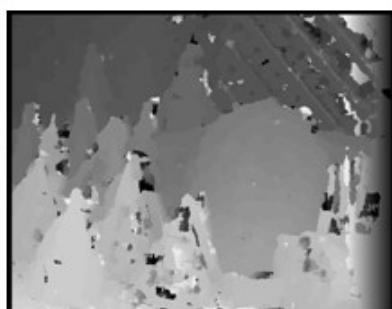
SHD s Census Transform



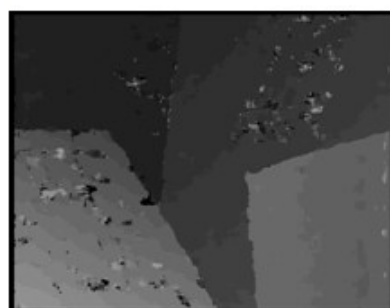
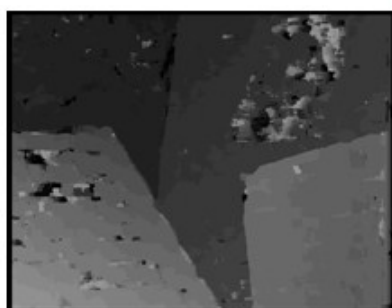
Teddy



Tsukuba



Cones



Venus

Obr. 6.7 Disparitní mapy vytvořené metodou SHD, vlevo bez použití census transformace, vpravo s jejím použitím

6.6 Srovnání

Pro každou testovanou metodu jsme vypočítali u nejlepších disparitních map (každé scény) průměrný počet vadných pixelů v procentech a průměrný čas pro všechny testy. Výsledky jsou obsaženy v tabulce 6.7. Nej kvalitnější disparitní mapy produkuje metoda ZSSD, ale za cenu největší časové náročnosti. Nejméně časově náročná je metoda SHD s použitím census transformovaných obrazů.

Metoda	Průměr vadných pixelů[%]	Průměrný čas [sec]
SSD	54,1	52,73
SAD	54,9	12,43
ZSSD	53,9	62,46
ZSAD	54,5	23,27
SHD	56,6	8,77
SHD(s CT)	56,8	8,69

Tabulka 6.7 Srovnání implementovaných metod

7 Závěr

V rámci této bakalářské práce byla nastudována a popsána problematika stereo korespondence. Jedná se o rozsáhlý problém, který doposud nebyl uspokojivě vyřešen. Postupně jsme se seznámili s problémy související s pořízením obrazu a jeho převodu do digitální podoby. Poté jsme se zaměřili na metody stereo korespondence. Rozdělili jsme je do skupin podle specifikace a popsali jejich principy, výhody a nevýhody.

Následně jsme naimplementovali vybrané metody s využitím knihovny OpenCV. Pro ohodnocení shody pixelů korespondujících obrazů jsme postupně použili metody SSD, SAD, ZSSD, ZSAD a SHD. Jednotlivé metody jsme dále podrobili sérii testů, díky kterým jsme zjistili jejich kvalitu a časovou náročnost. Z testů vyplynulo, že výsledná kvalita a časová náročnost je úzce spjata s volbou velikosti okénka(okolí pixelu) a maximální disparity, tj. maximální vzdálenost na řádku, pro kterou se mají hledat shodné pixely.

Pro zlepšení výsledků bychom mohli v programu implementovat metody, které se zabývají úpravou disparitních map. Tyhle metody dokážou vadným pixelům na základě jejich okolí vypočítat pravděpodobnou správnou hodnotu. Implementované metody pracují tak, že ze dvou vstupních obrazů se jeden vybere jako referenční, z kterého pak ohodnocujeme pixely v obraze korespondujícím. Proto další možný způsob jak dosáhnout lepších výsledků spočívá v tom, že algoritmus ohodnotí pixely z levého obrazu v pravém a následně pixely z pravého obrazu v levém. Tímto bychom lépe zobrazili zakryté body, ale za cenu větší časové náročnosti.

Algoritmy implementovaných metod by se taktéž dali urychlit. Jelikož metody pracují s každým řádkem obrazu zvlášť, možné urychlení programu by spočívalo v implementaci podpory vícejádrových procesorů. Pro dvoujádrový procesor bychom metody upravili tak, aby jeden procesor zpracovával liché řádky obrazu a druhý paralelně sudé.

8 Literatura

- [1] Hlaváč V., Šonka M.
Počítačové vidění. Grada 1992.
- [2] Hlaváč V., Sedláček M.
Zpracování signálů a obrazů, skriptum FEL ČVUT,
- [3] Holota R., Fiřt J.
Digitalizace a zpracování obrazu. s. 34-38. Plzeň, Západočeská univerzita, 2002,
ISBN 80-7082-917-6.
- [4] Gábor Blázsovits.
Interaktívna učebnica spracovania obrazu. 1 vyd, Bratislava, 2006,
ISBN 80-89186-08-4
- [5] Pavel Staněk
Stereoskopický senzor, Praha, 2005, diplomová práce
- [6] D. Scharstein and R. Szeliski.
A taxonomy and evaluation of dense two-frame stereo correspondence algorithms.
IJCV 2002
- [7] B. Cyganek a J. Paul Siebert.
An Introduction to 3D Computer Vision Techniques and Algorithms. 1.vyd, 2009
- [8] R. Zabih and J. Woodfill,
Non-parametric Local Transforms for Computing Visual Correspondence, ECCV 2004,
pp. 151-158
- [9] Young Ki Baik, Kyoung Mu Lee
Fast Census Transform-based Stereo Algorithm using SSE2
- [10] D. Scharstein a C. Pal.
Learning conditional random fields for stereo, 2007
- [11] H. Hirschmüller a D. Scharstein
Evaluation of cost functions for stereo matching, 2007